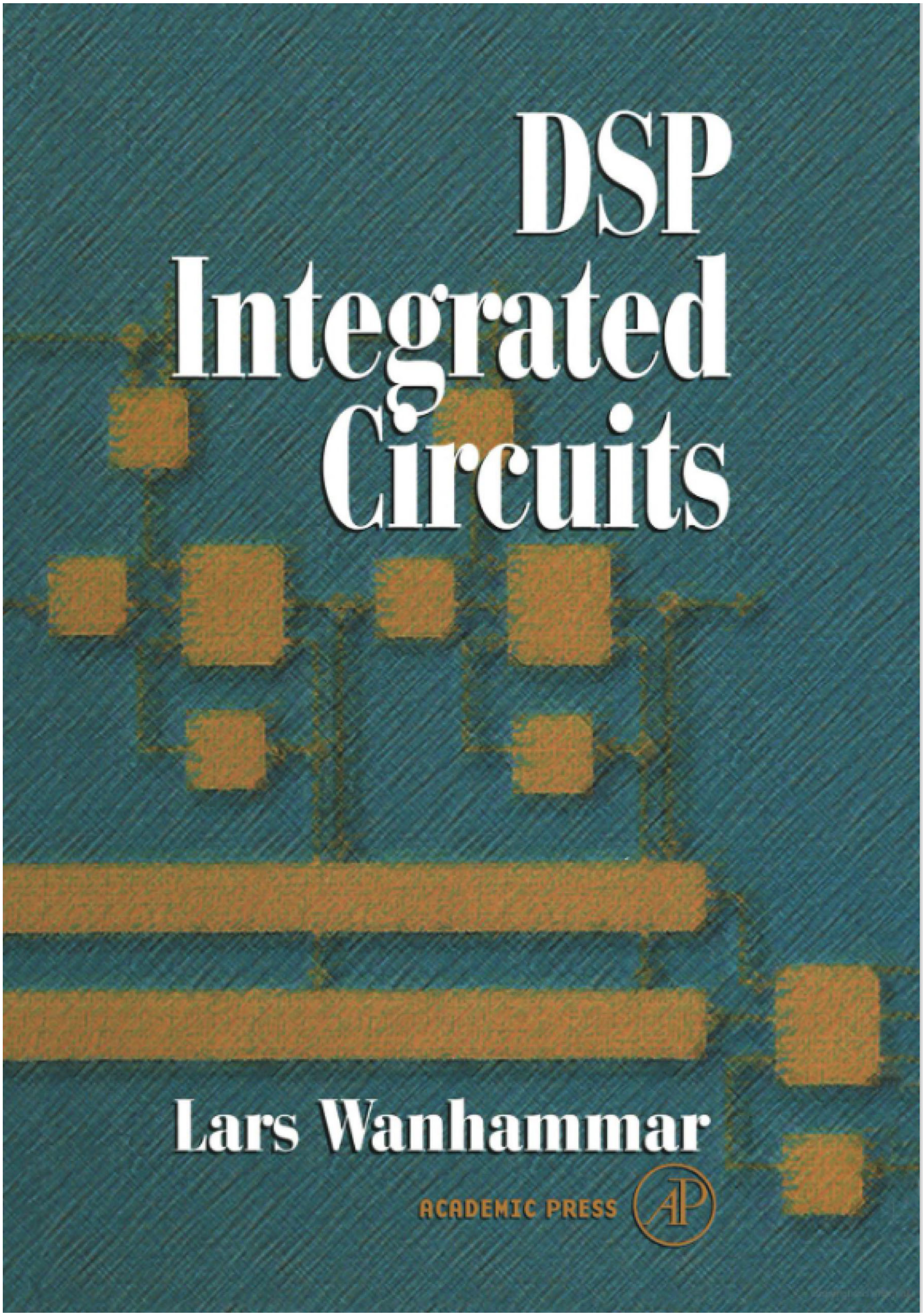
EXHIBIT A


DSP
Integrated
Circuits
Lars Wanhammar
ACADEMIC PRESS
AP

DSP INTEGRATED CIRCUITS

Lars Wanhammar
Linköping University

ACADEMIC PRESS
A Harcourt Science and Technology Company
San Diego San Francisco New York Boston London Sydney Tokyo

Academic Press Series in Engineering

Series Editor
J. David Irwin
Auburn University

Designed to bring together interdependent topics in electrical engineering, mechanical engineering, computer engineering, and manufacturing, the Academic Press Series in Engineering provides state-of-the-art handbooks, textbooks, and professional reference books for researchers, students, and engineers. This series provides readers with a comprehensive group of books essential for success in modern industry. A particular emphasis is given to the applications of cutting-edge research. Engineers, researchers, and students alike will find the Academic Press Series in Engineering to be an indispensable part of their design toolkit.

Published books in the series:
Industrial Controls and Manufacturing, 1999, E. Kamen
DSP Integrated Circuits, 1999, L. Wanhammar
Time Domain Electromagnetics, 1999, S.M. Rao
Single and Multi-Chip Microcontroller Interfacing, 1999, G.J. Lipovski

This book is printed on acid-free paper. ∞

Academic Press
A Harcourt Science and Technology Company
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
http://www.academicpress.com

Academic Press
Harcourt Place, 32 Jamestown Road, London NW1 7BY, UK
http://www.academicpress.com

Library of Congress Cataloging-in-Publication: 98-22149
ISBN: 0-12-734530-2

Printed in the United States of America
01 02 03 IP 9 8 7 6 5 4 3 2

of these assignments determine the communication requirement—i.e., communication channels and their bandwidth, etc. Hence, the minimum requirements are specified in these design steps. Therefore, to each static schedule corresponds a class of ideal multiprocessor architectures.

An *ideal DSP architecture* belongs to a class of architectures that implements the static schedule. An ideal architecture has processing elements that can execute the operations according to the schedule and is supported with appropriate communication channels and memories.

Note that there may be several architectures that implement a given schedule, and that a new class of architectures is obtained if the schedule is changed. Algorithms that require dynamic scheduling lead to architectures that either must handle worst-case situations or are optimized in a statistical sense. However, the execution time must be predictable since the sample period constraint must be met in hard real-time applications [13]. The latter type of architectures are therefore difficult to use.

8.4.1 Processing Elements

Processing elements (PEs) usually perform simple, memoryless mappings of the input values to a single output value. The arithmetic operations commonly used in DSP algorithms are

Add/sub, add/sub-and-shift

Multiply, multiply-and-accumulate

Vector product

Two-port adaptor

Butterfly

We will reserve the more general term *processor* to denote a PE with its internal memory and control circuitry. Hence, a processor is able to perform a task independently of other processors.

If several processing elements always operate on the same inputs, it may be advantageous to merge these into one PE with multiple inputs and outputs—for example, two-port adaptors and butterflies. Experience indicates that it is advantageous to use the largest operations possible (i.e., large PE granularity) since this tends to reduce the communication. However, flexibility in scheduling the operations is reduced and resource utilization may become poor if the operations chosen are too large. As always, a good trade-off is the best.

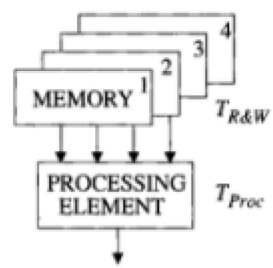


Figure 8.10 Processing element with multiple inputs

At this point it is interesting to note that the execution time for processing elements and the cycle time (read and write) for memories manufactured in the same technology are of the same order. Hence, to fully utilize a multiple-input processing element, as shown in Figure 8.10, one memory or memory port must be provided for each input and output value.